HOMOMORPHIC GENERATOR LEARNING: COMPOSITIONAL PDE NEURAL OPERATORS FOR MULTI-PHYSICS SURROGATES

Anonymous authors
Paper under double-blind review

ABSTRACT

PDE surrogates may be asked to predict systems whose physical mechanisms were observed separately, but never together, during training. Although the corresponding equation terms add, their solutions generally do not; a learned solution map therefore has no explicit rule for combining those observations. We propose Homomorphic Generator Learning (HGL), which learns a reusable generator for each mechanism and constructs an unseen system by adding its active generators. Each generator combines a role-constrained principal operator with a gated correction. Known symbols provide operator priors; otherwise, the symbols are learned from data. Mechanism coefficients are supplied by the equation or estimated from a short observed trajectory. A task-specific evaluator converts the combined generator into a rollout or steady-state solution. On unseen advective Fisher–KPP dynamics, HGL improves long-horizon prediction with either supplied or estimated coefficients relative to the evaluated solution-operator baselines. It also performs well as training coverage expands from individual mechanisms to partial combinations and when generators learned from different equation families are reused in a steady-state problem. These results support generator-level composition as an inductive bias for multi-physics generalization.

1 INTRODUCTION

Neural operators learn maps between fields, enabling repeated PDE queries without solving each instance from scratch (Kovachki et al., 2023). FNO (Li et al., 2021) and DeepONet (Lu et al., 2021) provide widely used solution-map architectures; DPOT (Hao et al., 2024) and Poseidon (Herde et al., 2024) extend pretraining across equation families. Equation-aware models also condition on governing terms (Zhu et al., 2025; Liu et al., 2024). A specific generalization question is whether a model can predict the simultaneous action of familiar mechanisms when their complete combination was absent from training. We examine this question for both time-dependent rollout and steady-state solution.

Consider a system in which transport moves a state, diffusion smooths it, and reaction changes it locally. Separate trajectories reveal how each term acts alone, but the solution of the joint equation is not obtained by summing those trajectories. Diffusion changes the state on which reaction acts at later times; in a steady boundary-value problem, transport and reaction alter the solution determined by diffusion. A model conditioned on a complete equation must infer this joint response from other configurations. Supplying coefficients or instantaneous tendency fields provides useful inputs, but does not impose an additive rule on the learned solution representation.

At the equation level, mechanism tendencies are added before the solution is computed (Evans, 2010). For discretized linear systems, the terms define generator components, and the combined evolution or resolvent depends on their sum (Pazy, 1983; Engel & Nagel, 2000). For nonlinear terms, the analogous object is a sum of state-dependent tendency maps. This suggests learning reusable terms and carrying out their interaction during numerical evaluation.

We introduce Homomorphic Generator Learning (HGL), which learns a generator for each mechanism and maps configuration merging to generator addition. This homomorphism makes each dictionary element reusable across systems. A principal operator encodes the mechanism's dynam-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ical role, and a gated correction accounts for bounded effects beyond that principal term. Known analytic symbols serve as priors; otherwise, a constrained symbol network learns them. Coefficients come from the equation or a short trajectory. An evolution or static evaluator then maps the combined generator to the requested solution. Figure 1 summarizes the workflow.

We test HGL on an unseen advective Fisher–KPP composition, a five-mechanism system under varying partial-combination coverage, and a steady convection–diffusion–reaction problem. These experiments test composition with both evaluator types. The principal terms encode dissipative, skew-adjoint, or local structure according to physical role (Reed & Simon, 1972; Shin et al., 2024).

Our contributions are: (1) an evaluation of unseen mechanism combinations in rollout and static solve; (2) a reusable, role-structured generator dictionary with an explicit additive composition law; and (3) a symbol-based parameterization with analytic priors where available, context-based coefficient estimation, and task-specific evaluators.

2 Related Work

Neural solution operators and representations. FNO (Li et al., 2021) models global interactions in Fourier space, while DeepONet (Lu et al., 2021) uses branch and trunk networks to learn function-to-function maps. CNO (Raonić et al., 2023), the Galerkin Transformer (Cao, 2021), and Transolver (Wu et al., 2024) provide convolutional, attention-based, and geometry-aware alternatives. PINO (Li et al., 2024) adds PDE-residual supervision, and PDNO (Shin et al., 2024) learns pseudodifferential symbols for a solution operator. These approaches fit within the broader neural-operator framework (Kovachki et al., 2023). HGL uses symbols to represent individual mechanism generators; a separate evaluator maps their sum to a solution.

Multi-equation pretraining and equation conditioning. DPOT (Hao et al., 2024) and Poseidon (Herde et al., 2024) learn transferable representations from multiple PDE families. PROSE-FD (Liu et al., 2024) and PROSE-PDE (Sun et al., 2025) combine field observations with equation information. PDEformer-1 (Ye et al., 2024) encodes symbolic and numerical PDE information in a computational graph; Unisolver (Zhou et al., 2025) embeds equation components as conditions, and equation-aware neural operators (Zhu et al., 2025) condition predictions on terms and coefficients. PDEBench (Takamoto et al., 2022) and The Well (Ohana et al., 2024) supply diverse evaluation systems. Our held-out-combination protocol tests whether individually observed mechanisms can be reused when their full combination was absent from training.

Context-based adaptation and identification. ICON (Yang et al., 2023; Yang & Osher, 2024) infers an operator from prompted input–output examples, whereas DISCO (Morel et al., 2025) uses a short trajectory to generate a compact evolution operator. PDE-FIND (Rudy et al., 2017) and its parametric extension (Rudy et al., 2019) identify governing terms or coefficient dependence from observations. HGL uses context for the narrower task of estimating coefficients over a fixed, mechanism-indexed generator dictionary. The same dictionary is used when those coefficients are supplied by the equation.

Compositional physical operators. Neural operator splitting (Serrano et al., 2026) searches a DISCO-derived operator dictionary against a test trajectory and composes selected dynamics through splitting. CompNO (Hmida et al., 2026) pretrains elementary-physics solution blocks and learns an adaptation block to aggregate their outputs for a target system. A physical-operator approach (Gopakumar et al., 2026) learns nonlinear terms separately, retains fixed finite-difference linear terms, and integrates their combination as a neural ODE. HGL learns coefficient-scaled, role-structured generators and applies either an evolution or a static inverse evaluator to their sum. For linear generators, semigroup and resolvent theory (Pazy, 1983; Engel & Nagel, 2000; Reed & Simon, 1972) and product formulas (Trotter, 1959; Strang, 1968; Hairer et al., 2006) motivate the distinction between composing generators and evaluating a solution. Pseudodifferential calculus (Hörmander, 2007; Taylor, 1981) motivates lower-order corrections for spatially varying media under suitable regularity assumptions.

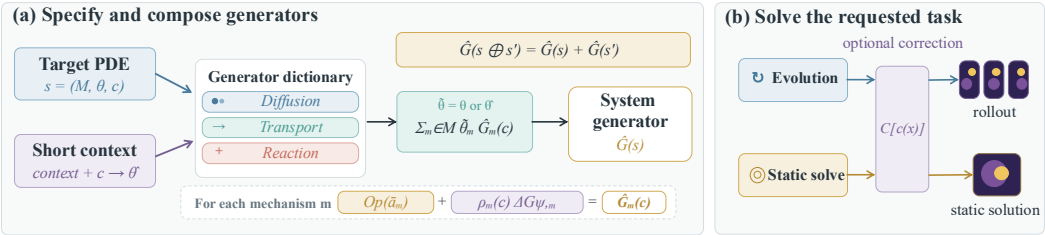


Figure 1: HGL pipeline. (a) PDE-supplied or context-estimated coefficients select and scale reusable mechanism generators. Each combines a principal operator with a gated correction; at fixed medium c, configuration merging corresponds to generator addition. (b) Evolution or static evaluation maps the combined generator to a rollout or static solution, with an optional medium correction.

3 Compositional Generalization Over PDE Mechanisms

Let $\mathcal{M}$ be a library of mechanisms and $s = (M, \theta, c)$ a system configuration. Here $M \subseteq \mathcal{M}$ is the active set, $\theta = (\theta_m)_{m\in\mathcal{M}}$ contains mechanism coefficients, and c specifies the medium through spatial coefficients, material fields, or geometry. In equation-conditioned tasks, M and θ are supplied. In trajectory-conditioned tasks, coefficients over the same library are inferred from context states; nonzero entries select active mechanisms. For configurations $s = (M, \theta, c)$ and $s' = (M', \theta', c)$ at fixed c, merging $s \oplus s'$ takes the union of active sets and adds overlapping coefficients. On the ambient coefficient space, these configurations form a commutative semigroup. Experimental coefficient ranges need not be closed under this operation. We test configurations whose complete mechanism set never appeared in training, although each component did.

The withheld object is a combination, not a mechanism absent from the dictionary. Training for the three-mechanism rollout contains only singletons, while testing uses their triple. The five-mechanism study progressively adds partial combinations but withholds the five-term system; the steady problem withholds the joint transport–reaction pairing with diffusion. This protocol separates reuse of known terms from discovering a new term, although coefficients may also shift at test time.

Time-dependent inputs contain context states $u_{t-k+1:t}$ and target future states $u_{t+1:t+R}$, with context length k and rollout length R. Static inputs contain a source and coefficient fields, with a steady-state solution as target. The predictor receives the available configuration fields; when coefficients are withheld, it estimates them from the context. Both tasks use the same mechanism representation. For a nonlinear PDE, a generator is a state-dependent tendency map, $G_s(u; c) = \sum_{m\in M} \theta_m G_m(u; c)$; for a discretized linear PDE, each G_m is a matrix. Additivity refers to the governing tendency at a fixed state and medium, not to the resulting trajectories or steady solutions.

For discretized linear generators A and B, evolution and static solves involve $e^{\Delta t(A+B)}$ and $(\mu - A - B)^{-1}$, respectively, where Δt is a time step and μ a resolvent parameter. Neither response is generally a sum of separate responses. When A and B do not commute, sequential evolution also differs from exact evolution through commutator terms $[A, B] = AB - BA$ (Hairer et al., 2006). For nonlinear mechanisms, the evaluator integrates the summed state-dependent tendency, retaining the same composition rule without invoking a linear operator exponential.

At fixed medium, coefficients scale reusable dictionary entries, while the evaluator acts on their sum. A generator remains mechanism-specific even when its effect on a trajectory or static solution depends on the other active terms.

4 Method

HGL adds reusable mechanism generators, then evaluates their sum for the requested task. Principal terms encode mechanism roles; gated residuals and optional medium-dependent corrections capture additional effects. Figure 1 shows the complete construction.

4.1 HOMOMORPHIC MECHANISM COMPOSITION

Let $\widehat{G}_m(c)$ denote the learnable generator of mechanism m under medium condition c, and let $\widehat{G}(s)$ denote the combined generator for configuration s. For $s = (M, \theta, c)$, the model constructs

$$\widehat{G}(s) = \sum_{m \in M} \theta_m \widehat{G}_m(c). \tag{1}$$

Equation 1 uses supplied coefficients θ or context-estimated coefficients $\widehat{\theta} = E_{\mathrm{sym}}(u_{\mathrm{ctx}})$. The latter select and scale entries in the same dictionary; Section 5 uses a parameter-free spectral estimator. Each dictionary entry is decomposed into a principal mechanism operator and a bounded correction. Let $\bar{a}_m(x, \xi; c)$ denote the principal symbol of mechanism m, so that $\mathrm{Op}(\bar{a}_m)$ is its principal operator. Let $\Delta G_{\psi,m}(c)$ denote a learnable correction with parameters ψ, and let $\rho_m(c) \in [0, 1]$ denote its configuration-dependent residual weight. We write

$$\widehat{G}_m(c) = \mathrm{Op}\big(\bar{a}_m(\cdot, \cdot; c)\big) + \rho_m(c)\, \Delta G_{\psi,m}(c). \tag{2}$$

The principal symbol sets the leading spatial order. An analytic unit-strength symbol supplies $\bar{a}_m$ when available; otherwise, a role-constrained symbol network learns it. The coefficient θ_m or $\widehat{\theta}_m$ scales the mechanism, while $\rho_m(c)$ gates the bounded correction. Write $\Gamma(s) = \widehat{G}(s)$. For configurations that share the same medium condition and can be merged, Γ satisfies

$$\Gamma(s \oplus s') = \Gamma(s) + \Gamma(s'). \tag{3}$$

For fixed c, Equation 3 makes the dictionary map a semigroup homomorphism. Let $\Phi(s)$ denote the prediction map, Eval a task evaluator, and $\mathcal{C}_c$ an optional lower-order medium-interaction map. Schematically,

$$\Phi(s) = \mathcal{C}_c \circ \mathrm{Eval}[\widehat{G}(s)]. \tag{4}$$

Let $\mathrm{Eval}_{\mathrm{evo}}$ and $\mathrm{Eval}_{\mathrm{stat}}$ denote the evolution and static evaluators, respectively. For linear grid generators, splitting and preconditioned iteration approximate the exponential and inverse maps (Trotter, 1959; Strang, 1968; Hairer et al., 2006; Pazy, 1983; Hörmander, 2007):

$$\mathrm{Eval}_{\mathrm{evo}}[G] \approx e^{\Delta t G}, \qquad \mathrm{Eval}_{\mathrm{stat}}[G] \approx (\mu - G)^{-1}. \tag{5}$$

For nonlinear tendencies, $\mathrm{Eval}_{\mathrm{evo}}$ integrates the summed vector field; the static evaluator, when applicable, solves the corresponding residual equation. The additive dictionary is shared across configurations, while the evaluator follows the numerical structure of the target task. The medium correction may be integrated into its iterations (Appendix C).

4.2 STRUCTURED MECHANISM GENERATORS

The principal terms belong to three role-specific families. A finite-grid diffusion symbol has a nonpositive Hermitian part; the unit-strength Laplacian has symbol $-|\xi|^2$ and damps high-frequency modes. Constant-velocity transport has symbol $-i(v \cdot \xi)$, a skew-Hermitian principal part that changes phase without changing amplitude. Reaction and forcing are zero-order local maps, such as $f(u, x)$, with no spatial mixing in the principal term. Each receives a bounded correction as in Equation 2. These constraints apply to principal components; full operator properties require domain and discretization assumptions (Appendix A) (Evans, 2010; Pazy, 1983; Reed & Simon, 1972).

4.3 PSEUDODIFFERENTIAL GENERATOR PARAMETERIZATION

Pseudodifferential symbol networks parameterize the principal terms on discrete grids, with nonpositive Hermitian, skew-Hermitian, or local structure as appropriate. PDNO (Shin et al., 2024) uses related symbol layers for solution-operator learning; here they parameterize reusable mechanism terms.

For a nonuniform coefficient field $c(x)$, let $M_c u = c\, u$. If $\mathrm{Op}(a_m)$ has order p, standard symbol calculus gives $[\mathrm{Op}(a_m), M_c] \in \Psi^{p-1}$, the order-$(p-1)$ class, with leading symbol controlled by $\partial_\xi a_m \cdot \nabla c$ (Hörmander, 2007; Taylor, 1981). HGL represents this interaction by the bounded lower-order map $\mathcal{C}_c$ in Equation 4; its parameterization and discrete counterpart are given in Appendix B.

On a finite grid, diffusion and transport symbols are projected onto non-positive Hermitian and skew-Hermitian classes, while reaction and source branches use pointwise maps. The additional term in the optional medium-interaction map $\mathcal{C}_c$ is gated by $|\nabla c|$ and vanishes for a spatially uniform medium.

5 EXPERIMENTS

We test whether HGL can reuse mechanisms that appear separately in training to predict systems formed by unseen combinations. The experiments cover time-dependent rollouts and steady-state boundary-value problems. We also evaluate context-based coefficient estimation and data-driven generator learning when analytic symbols are unavailable.

5.1 EVALUATION SETUP

Let $\widehat{u}$ denote the model prediction, u the reference solution, $\|\cdot\|_2$ the discrete L^2 norm, and $\epsilon > 0$ a small constant that prevents division by zero. The primary metric is

$$e_{\mathrm{rel}}(\widehat{u}, u) = \frac{\|\widehat{u} - u\|_2}{\|u\|_2 + \epsilon}. \tag{6}$$

Rollout horizons are reported in forecast steps. Unless otherwise stated, we first take the median over test samples for each training run and then the median across three random seeds. Within each protocol, all models receive the same state history and system inputs, use the same data splits and target times, and have comparable parameter counts. In the primary Fisher–KPP comparison, HGL and all baselines receive equation-specified mechanism coefficients. The fitted variant estimates these coefficients from four context frames. Training details are provided in Appendix D.

The time-dependent composition experiments compare U-Net (Ronneberger et al., 2015), DeepONet (Lu et al., 2021), DPOT (Hao et al., 2024), CNO (Raonić et al., 2023), FNO (Li et al., 2021), Galerkin Transformer (Cao, 2021), and Transolver (Wu et al., 2024). The steady-state composition experiment compares parameter-matched FNO and Transolver models.

5.2 UNSEEN MECHANISM COMPOSITIONS

Generalization to an unseen advective Fisher–KPP system. We first evaluate generalization from single-mechanism training to an unseen three-mechanism system. The test system is the Fisher–KPP equation with constant transport on the two-dimensional periodic domain $\mathbb{T}^2$ (Fisher, 1937):

$$\partial_t u(x,t) = -\mathbf{v}\cdot\nabla u(x,t) + \nu\Delta u(x,t) + \rho u(x,t)\big(1 - u(x,t)\big), \qquad x \in \mathbb{T}^2. \tag{7}$$

Here $u : \mathbb{T}^2 \times \mathbb{R}_+ \to \mathbb{R}$ is the state field and $t \geq 0$ is time. The vector $\mathbf{v} = (v_x, v_y) \in \mathbb{R}^2$ is the transport velocity, $\nu > 0$ is the diffusion coefficient, and $\rho \geq 0$ is the logistic reaction coefficient. Each training trajectory contains only transport, diffusion, or reaction, whereas each test trajectory contains all three mechanisms. The test set includes coefficient combinations within the corresponding single-mechanism training ranges, together with extrapolation cases whose diffusion coefficients lie below the training range. Let $\theta = (v_x, v_y, \nu, \rho)$ denote the mechanism coefficient vector and $\widehat{\theta}$ its estimate. HGL uses the three unit-strength operator symbols as the principal terms in Equation 2, then adds their generators to construct the three-mechanism system.

For the fitted variant, a parameter-free spectral estimator uses four context frames to identify transport and diffusion coefficients, and a spatial logistic balance estimates reaction strength (Rudy et al., 2017; 2019). The regression bands and estimator are specified in Appendix D.

The estimator fits transport from changes in the phase of low-frequency Fourier coefficients and diffusion from frequency-dependent changes in their amplitude. On the periodic domain, spatial averaging eliminates the transport and diffusion contributions, so the change in the state mean can be regressed against the averaged logistic term to estimate reaction strength. These estimates scale the same generator entries used when coefficients are supplied.

Table 1: Median relative L^2 rollout error (three seeds) on the unseen advective Fisher–KPP composition after single-mechanism training. HGL (fitted $\widehat{\theta}$) estimates coefficients from four context frames; other rows use equation-specified coefficients. Parameters are in millions; bold denotes the lowest error per horizon.

		Prediction steps				
Model	Params	10	20	40	60	80
HGL	2.03	$\mathbf{7.7\times10^{-5}}$	$\mathbf{1.4\times10^{-4}}$	$\mathbf{2.4\times10^{-4}}$	$\mathbf{3.1\times10^{-4}}$	$\mathbf{3.6\times10^{-4}}$
HGL (fitted $\widehat{\theta}$)	2.03	0.0055	0.0100	0.0173	0.0224	0.0257
FNO	2.08	0.539	0.757	0.813	0.710	0.566
DPOT	2.22	0.559	0.767	0.820	0.711	0.555
Transolver	2.04	0.578	0.792	0.822	0.727	0.542
U-Net	2.05	0.562	0.767	0.769	0.648	0.481
Galerkin Transformer	2.04	0.578	0.789	0.804	0.676	0.469
DeepONet	2.08	0.498	0.604	0.572	0.541	0.544
CNO	2.06	0.584	0.807	0.844	0.732	0.548

Table 1 shows that HGL with equation-specified coefficients has the lowest error at every horizon; its context-fitted variant also remains below the evaluated baselines. The same ordering holds in low-diffusion extrapolation.

Mechanism and inference diagnostics. We test prior-informed baselines and generator composition in Tables 2 and 3; coefficient-estimation sensitivity is reported in Appendix Table 7. Direct baselines in Table 2 receive equation-specified coefficients and the three instantaneous mechanism-tendency fields.

Both HGL variants retain lower long-horizon errors than the prior-informed baselines. The composition ablation holds the mechanism generators, symbols, and evaluator fixed across the additive, nonlinear, and residual nonlinear variants.

Single-mechanism training contains no joint activations, so a nonlinear mixer must extrapolate its response to the three-mechanism input; addition specifies the combined generator directly. The residual nonlinear mixer fits seen systems but raises unseen step-80 error by about a factor of 15. The unconstrained-symbol variant performs slightly better than HGL with type-consistent initialization, so the ablation supports additive composition more directly than the symbol constraints.

Clean coefficient estimates change little with context length; observation noise increases coefficient and rollout errors. Protocol details appear in Appendix D.

Table 2: Long-horizon median relative L^2 errors with known mechanism symbols (three seeds). Direct baselines receive equation-specified coefficients and the three instantaneous mechanism-tendency fields; HGL (fitted $\widehat{\theta}$) estimates coefficients from four context frames, while HGL uses equation-specified coefficients. Parameter counts are in millions; column minima are bold.

		Coefficient interpolation		Low diffusion	
Model	Params	60 steps	80 steps	60 steps	80 steps
HGL	2.03	$\mathbf{3.1\times10^{-4}}$	$\mathbf{3.6\times10^{-4}}$	$\mathbf{3.1\times10^{-4}}$	$\mathbf{3.5\times10^{-4}}$
HGL (fitted $\widehat{\theta}$)	2.03	0.0224	0.0257	0.0211	0.0230
DeepONet	2.09	0.173	0.390	0.436	3.933
U-Net	2.05	0.176	0.340	0.204	0.398
CNO	2.07	0.200	0.215	0.241	0.252
FNO	2.08	0.341	0.380	0.480	0.484
DPOT	1.54	0.478	0.516	0.570	0.576
Transolver	2.04	0.735	0.837	0.995	1.065

Table 3: Composition and symbol ablations with one seed and 200 training steps. Seen-system validation is measured by MSE; unseen three-mechanism errors are median relative L^2 errors. Column minima are bold.

	Seen-system val.	Unseen three-mechanism system		
Model	MSE	10 steps	60 steps	80 steps
HGL (additive)	$\mathbf{1.31 \times 10^{-8}}$	7.74×10^{-5}	3.09×10^{-4}	3.55×10^{-4}
Nonlinear composition	1.89×10^{-1}	5.09×10^{-1}	5.39×10^{-1}	4.21×10^{-1}
Residual nonlinear composition	1.05×10^{-5}	1.05×10^{-3}	4.58×10^{-3}	5.27×10^{-3}
Unconstrained symbols	2.07×10^{-8}	$\mathbf{7.01 \times 10^{-5}}$	$\mathbf{2.39 \times 10^{-4}}$	$\mathbf{2.66 \times 10^{-4}}$
Learned symbols	1.82×10^{-1}	5.25×10^{-1}	5.22×10^{-1}	4.06×10^{-1}

Partial-combination coverage. We use a two-dimensional PDEBench setting (Takamoto et al., 2022) with five mechanisms: linear transport, isotropic diffusion, nonlinear advection, reaction, and external forcing. Training systems contain at most one, two, three, or four mechanisms. The held-out system contains all five, and this full composition never appears during training. HGL uses one fixed parameterization across coverage levels: linear transport and diffusion use known operator symbols, while the other three mechanisms are learned from data.

The coverage levels contain 5, 15, 25, and 30 active sets; training trajectories and optimization steps (200, 600, 1000, and 1200) increase together with coverage.

HGL has the lowest error in all 16 settings of Table 4. Its 60-step error falls from 0.385 with single-mechanism training to 0.179 with up to three mechanisms and is 0.193 with up to four. In Figure 2a, HGL preserves the locations and amplitudes of high-value structures, while baseline predictions show amplitude decay, displacement, or high-frequency oscillations. Additional trajectories appear in Appendix Figure 3.

5.3 Steady Convection–Diffusion–Reaction Equations

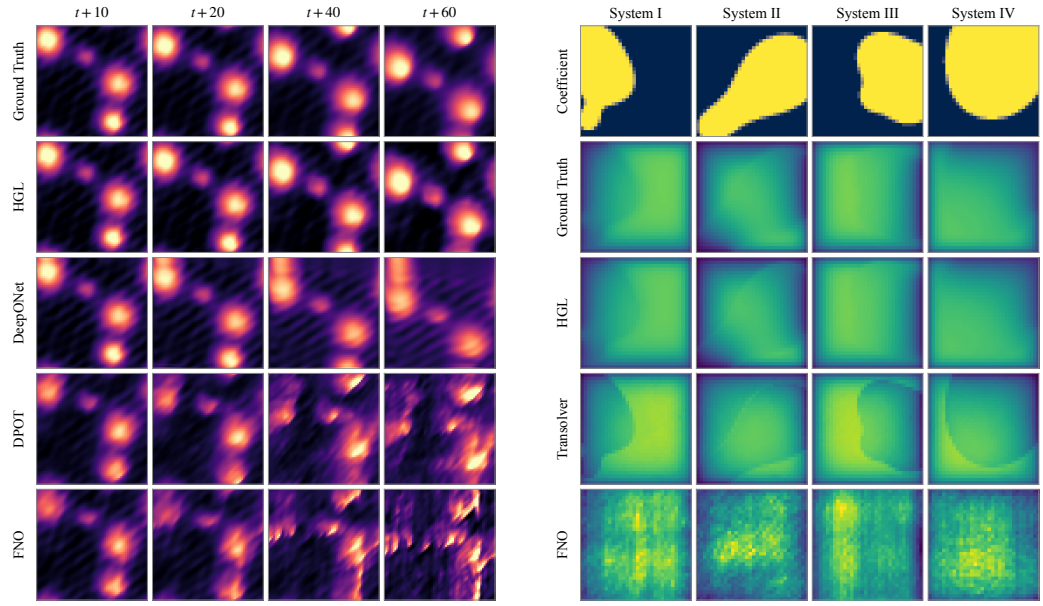
This experiment asks whether an unseen three-mechanism operator can be assembled by generator addition and evaluated through a steady-state inverse solve. The test problem is a standard steady convection–diffusion–reaction boundary-value problem (Deuring, 2016). Let $\Omega = (0,1)^2$ denote the spatial domain, $x \in \Omega$ a spatial coordinate, and $\partial\Omega$ the domain boundary:

$$\mathcal{L}_{\nu,\mathbf{b},r} u = -\nabla \cdot \big(\nu(x)\nabla u(x)\big) + \mathbf{b}\cdot\nabla u(x) + r u(x) = f(x), \qquad u|_{\partial\Omega} = 0, \tag{8}$$

Here $u : \Omega \to \mathbb{R}$ is the steady-state solution and $\nu : \Omega \to \mathbb{R}_+$ is the heterogeneous diffusion coefficient. The constant transport velocity is $\mathbf{b} = (b_x, b_y) \in \mathbb{R}^2$, the reaction coefficient satisfies $r \geq 0$, and the source is $f \equiv 1$. The operator $\mathcal{L}_{\nu,\mathbf{b},r}$ is the sum of diffusion, transport, and reaction terms, with generator $G_{\nu,\mathbf{b},r} = -\mathcal{L}_{\nu,\mathbf{b},r}$. The static evaluator computes $(-G_{\nu,\mathbf{b},r})^{-1} f = \mathcal{L}^{-1}_{\nu,\mathbf{b},r} f$.

Table 4: Median relative L^2 rollout error on the unseen five-mechanism system by maximum training-composition size. Best baseline is the lowest error among seven direct solution-operator models at each horizon; bold denotes the lower value within each pair.

Max. mechanisms	Model	10 steps	20 steps	40 steps	60 steps
1	HGL	**0.096**	**0.174**	**0.299**	**0.385**
	Best baseline	0.251	0.428	0.603	0.662
2	HGL	**0.056**	**0.110**	**0.216**	**0.318**
	Best baseline	0.120	0.214	0.357	0.429
3	HGL	**0.037**	**0.067**	**0.124**	**0.179**
	Best baseline	0.087	0.165	0.302	0.389
4	HGL	**0.035**	**0.069**	**0.128**	**0.193**
	Best baseline	0.067	0.127	0.229	0.308



(a) Time-dependent five-mechanism composition.

(b) Steady convection–diffusion–reaction equations.

Figure 2: Predictions on unseen compositions. (a) Five-mechanism rollout at four forecast steps. (b) Steady convection–diffusion–reaction solutions for four coefficient fields. Time-dependent panels share a color scale; each steady-state column uses a shared scale.

The diffusion-coefficient fields are taken from the PDEBench Darcy data (Takamoto et al., 2022). The training set covers three equation subfamilies: heterogeneous diffusion ($\mathbf{b} = 0, r = 0$), convection–diffusion ($\mathbf{b} \neq 0, r = 0$), and screened Poisson ($\mathbf{b} = 0, r > 0$). Three-mechanism equations containing both transport and reaction ($\mathbf{b} \neq 0, r > 0$) appear only at test time. All learned models receive the same ν, $\mathbf{b}$, and r.

Reference solutions use a 32×32 interior grid with disjoint training, validation, and test diffusion fields.

HGL forms $G_{\nu,\mathbf{b},r}$ by adding the three mechanism operators in Equation 8. The static evaluator performs 24 preconditioned residual updates. Its constant-coefficient preconditioner is inverted by a discrete sine transform after exponential conjugation removes transport; a learned local correction uses the iterate, residual, and coefficient fields. The update and preconditioner construction are shared across equation families (Appendix C).

Table 5 shows that HGL obtains the lowest error on the three-mechanism test equation and all three training subfamilies. Figure 2b shows that HGL reconstructs the spatial structure of the ground-truth solutions for four heterogeneous coefficient fields. Transolver distorts the boundaries of high-value regions, while FNO produces local oscillations.

Table 5: Median relative L^2 error (three seeds, physical units) on the unseen full convection–diffusion–reaction equation and three seen subfamilies. Parameters are in millions; bold denotes the lowest error per column.

		Unseen	Seen training families		
Model	Params	Full equation	Darcy	Conv.–diff.	Screened Poisson
HGL	0.407	$\mathbf{2.55 \times 10^{-2}}$	$\mathbf{5.58 \times 10^{-3}}$	$\mathbf{8.42 \times 10^{-3}}$	$\mathbf{7.13 \times 10^{-3}}$
Transolver	0.450	1.11×10^{-1}	6.34×10^{-2}	7.98×10^{-2}	7.52×10^{-2}
FNO	0.463	2.76×10^{-1}	2.83×10^{-1}	2.70×10^{-1}	2.36×10^{-1}

Table 6: Median relative L^2 errors on The Well over three runs. Parameters are in millions; bold marks the lowest error per dataset and protocol.

Dataset	Model	Params	Traj. val.	Param. interp.	Param. extrap.
Active matter	HGL	3.53	**1.010**	**0.773**	0.640
Active matter	TANTE	4.23	1.640	0.878	**0.611**
Active matter	AFNO	5.64	2.120	0.802	0.614
Active matter	FNO	3.66	1.890	1.021	1.000
Active matter	U-Net-C	4.04	3.950	1.179	1.035
Turbulent layer	HGL	3.53	**0.430**	0.488	**0.236**
Turbulent layer	TANTE	4.23	0.500	0.515	0.251
Turbulent layer	AFNO	5.64	0.466	0.485	0.250
Turbulent layer	FNO	3.66	0.459	**0.481**	0.255
Turbulent layer	U-Net-C	4.04	0.560	0.565	0.476
Helmholtz	HGL	3.53	**0.036**	**0.042**	0.107
Helmholtz	TANTE	4.23	0.094	0.055	**0.096**
Helmholtz	AFNO	5.64	0.079	0.075	0.152
Helmholtz	FNO	3.66	0.190	0.155	0.241
Helmholtz	U-Net-C	4.04	0.468	0.388	0.576

5.4 Learned Symbols in The Well

The Well (Ohana et al., 2024) supplies three multi-field systems without analytic mechanism symbols: active matter, a turbulent radiative layer, and the Helmholtz staircase. The varied parameters are dipole and alignment strengths, cooling time, and acoustic frequency, respectively. We evaluate trajectory validation and unseen parameter settings within and beyond the training ranges. HGL, TANTE (Wu et al., 2026), AFNO (Guibas et al., 2022), FNO, and a U-Net use the same four-frame context, eight-step target, data splits, and validation-based checkpoint selection. Appendix E gives the parameter ranges.

HGL has the lowest error in six of nine settings (Table 6). The exceptions are active-matter and Helmholtz extrapolation and turbulent-layer interpolation. This result supports learned symbols across these datasets while delimiting their advantage under parameter shifts.

6 Conclusion

HGL composes reusable mechanism generators by addition and evaluates the resulting operator for rollout or steady-state solution. Across advective Fisher–KPP dynamics, a five-mechanism system, and steady convection–diffusion–reaction equations, the method transfers to combinations absent from training. The additive-composition ablation supports the role of this inductive bias. Future work includes operator-level guarantees for broader nonlinear mechanisms and adaptation to mechanisms absent from the training library.

Reproducibility statement

An anonymous reproducibility artifact is available on OSF. The MIT-licensed archive contains the HGL and baseline implementations, software requirements, table-specific training commands, and scripts that generate synthetic trajectories or acquire external datasets. The Fisher–KPP and five-mechanism data are generated and cached by the experiment scripts; the steady CDR preparation script downloads the cited PDEBench Darcy fields and constructs the equation families used here. Appendix D specifies data splits, coefficient estimation, and training budgets; Appendix C details the static evaluator.

ETHICS STATEMENT

The experiments use synthetic PDE trajectories and publicly released scientific simulation data from PDEBench and The Well. They involve no human participants or personal data. Data sources and preparation are documented in the cited dataset papers, Appendix D, and the reproducibility artifact.

AI USE STATEMENT

Large language models assisted with drafting and polishing explanatory prose, clarifying the interpretation of reported results, discovering related work, and checking reference metadata. The authors are responsible for the final text, citations, and scientific claims.

REFERENCES

Shuhao Cao. Choose a transformer: Fourier or Galerkin. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24924–24940, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/d0921d442ee91b896ad95059d13df618-Abstract.html.

Paul Deuring. A finite element–finite volume discretization of convection–diffusion–reaction equations with nonhomogeneous mixed boundary conditions: Error estimates. *Numerical Methods for Partial Differential Equations*, 32(6):1591–1621, 2016. doi: 10.1002/num.22064.

Klaus-Jochen Engel and Rainer Nagel. *One-Parameter Semigroups for Linear Evolution Equations*, volume 194 of *Graduate Texts in Mathematics*. Springer, New York, 2000. doi: 10.1007/b97696.

Lawrence C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2 edition, 2010. doi: 10.1090/gsm/019.

R. A. Fisher. The wave of advance of advantageous genes. *Annals of Eugenics*, 7(4):355–369, 1937. doi: 10.1111/j.1469-1809.1937.tb02153.x.

Vignesh Gopakumar, Ander Gray, Daniel Giles, Lorenzo Zanisi, Matt J. Kusner, Timo Betcke, Stanislas Pamela, and Marc Peter Deisenroth. Learning physical operators using neural operators. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 3223–3231. PMLR, 2026. URL https://proceedings.mlr.press/v300/gopakumar26a.html.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Efficient token mixing for transformers via adaptive Fourier neural operators. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=EXHG-A3jlM.

Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, volume 31 of *Springer Series in Computational Mathematics*. Springer, 2 edition, 2006. doi: 10.1007/3-540-30666-8.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17616–17635. PMLR, 2024. URL https://proceedings.mlr.press/v235/hao24d.html.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 72525–72624, 2024. doi: 10.52202/079017-2311.

Hamda Hmida, Hsiu-Wen Chang, and Youssef Mesri. Compositional neural operators for multi-dimensional fluid dynamics. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VFLwvtTwWF.

Lars Hörmander. *The Analysis of Linear Partial Differential Operators III: Pseudo-Differential Operators*. Classics in Mathematics. Springer, Berlin, Heidelberg, 2007. doi: 10.1007/978-3-540-49938-1.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023. URL https://jmlr.org/papers/v24/21-1524.html.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.08895.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/IMS Journal of Data Science*, 1(3):1–27, 2024. doi: 10.1145/3648506.

Yuxuan Liu, Jingmin Sun, Xinjie He, Griffin Pinney, Zecheng Zhang, and Hayden Schaeffer. PROSE-FD: A multimodal PDE foundation model for learning multiple operators for forecasting fluid dynamics. *arXiv preprint arXiv:2409.09811*, 2024. URL https://arxiv.org/abs/2409.09811.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5.

Rudy Morel, Jiequn Han, and Edouard Oyallon. DISCO: Learning to DISCover an evolution operator for multi-physics-agnostic prediction. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 44750–44774. PMLR, 2025. URL https://proceedings.mlr.press/v267/morel25a.html.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam Parker, Miles Cranmer, and Shirley Ho. The Well: a large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 44989–45037, 2024. doi: 10.52202/079017-1430.

Amnon Pazy. *Semigroups of Linear Operators and Applications to Partial Differential Equations*, volume 44 of *Applied Mathematical Sciences*. Springer, New York, 1983. doi: 10.1007/978-1-4612-5561-1.

Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of PDEs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 77187–77200, 2023. doi: 10.52202/075280-3376.

Michael Reed and Barry Simon. *Methods of Modern Mathematical Physics I: Functional Analysis*. Academic Press, New York, 1972. doi: 10.1016/B978-0-12-585001-8.X5001-6.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351 of *Lecture Notes in Computer Science*, pp. 234–241. Springer, 2015. doi: 10.1007/978-3-319-24574-4_28.

Samuel Rudy, Alessandro Alla, Steven L. Brunton, and J. Nathan Kutz. Data-driven identification of parametric partial differential equations. *SIAM Journal on Applied Dynamical Systems*, 18(2):643–660, 2019. doi: 10.1137/18M1191944.

Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017. doi: 10.1126/sciadv.1602614.

Louis Serrano, Jiequn Han, Edouard Oyallon, Shirley Ho, and Rudy Morel. Test-time generalization for physics through neural operator splitting. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/62469.

Jin Young Shin, Jae Yong Lee, and Hyung Ju Hwang. Pseudo-differential neural operator: Generalized Fourier neural operator for learning solution operators of partial differential equations. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=805jKZ0Gqf.

Gilbert Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968. doi: 10.1137/0705041.

Jingmin Sun, Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. Towards a foundation model for partial differential equations: Multioperator learning and extrapolation. *Physical Review E*, 111 (3):035304, 2025. doi: 10.1103/PhysRevE.111.035304.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1596–1611, 2022. doi: 10.52202/068431-0117.

Michael E. Taylor. *Pseudodifferential Operators*, volume 34 of *Princeton Mathematical Series*. Princeton University Press, Princeton, NJ, 1981.

H. F. Trotter. On the product of semi-groups of operators. *Proceedings of the American Mathematical Society*, 10(4):545–551, 1959. doi: 10.1090/S0002-9939-1959-0108732-6.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53681–53705. PMLR, 2024. URL https://proceedings.mlr.press/v235/wu24r.html.

Zhikai Wu, Sifan Wang, Shiyang Zhang, Sizhuang He, Min Zhu, Anran Jiao, Lu Lu, and David van Dijk. TANTE: Time-adaptive operator learning via neural Taylor expansion. *Journal of Computational Physics*, 562:115041, 2026. doi: 10.1016/j.jcp.2026.115041.

Liu Yang and Stanley J. Osher. PDE generalization of in-context operator networks: A study on 1D scalar nonlinear conservation laws. *Journal of Computational Physics*, 519:113379, 2024. doi: 10.1016/j.jcp.2024.113379.

Liu Yang, Siting Liu, Tingwei Meng, and Stanley J. Osher. In-context operator learning with data prompts for differential equation problems. *Proceedings of the National Academy of Sciences*, 120(39):e2310142120, 2023. doi: 10.1073/pnas.2310142120.

Zhanhong Ye, Xiang Huang, Leheng Chen, Zining Liu, Bingyang Wu, Hongsheng Liu, Zidong Wang, and Bin Dong. PDEformer-1: A foundation model for one-dimensional partial differential equations. *arXiv preprint arXiv:2407.06664*, 2024. URL https://arxiv.org/abs/2407.06664.

Hang Zhou, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. Unisolver: PDE-conditional transformers towards universal neural PDE solvers. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79061–79088. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhou25r.html.

Qian-Ze Zhu, Paul Raccuglia, and Michael P. Brenner. Generalizing PDE emulation with equation-aware neural operators. *arXiv preprint arXiv:2511.09729*, 2025. URL https://arxiv.org/abs/2511.09729.

A Algebraic and Operator Setting

Using the notation of Section 3, extend coefficients of inactive mechanisms by zero. For configurations $s = (M, \theta, c)$ and $s' = (M', \theta', c)$ with a common medium condition, define

$$s \oplus s' = (M \cup M', \theta + \theta', c), \tag{9}$$

where coefficient addition is componentwise. Applying Equation 1 gives

$$\begin{aligned} \Gamma(s \oplus s') &= \sum_{m \in \mathcal{M}} (\theta_m + \theta'_m) \widehat{G}_m(c) \\ &= \Gamma(s) + \Gamma(s'). \end{aligned} \tag{10}$$

Thus, for each fixed c, Γ is a homomorphism from configuration merging to generator addition. Configurations with different medium conditions are represented by the conditional family $c \mapsto \widehat{G}_m(c)$ rather than by the same binary operation.

For bounded generators A and B, the first nontrivial term in the BCH expansion is

$$\log\left(e^{\Delta t A} e^{\Delta t B}\right) = \Delta t(A + B) + \frac{\Delta t^2}{2}[A, B] + O(\Delta t^3) \tag{11}$$

where $[A, B] = AB - BA$, Δt is the time step, and $O(\Delta t^3)$ collects higher-order terms (Hairer et al., 2006). For unbounded generators, the same expansion requires a common invariant domain and the corresponding operator-theoretic conditions.

The experiments use bounded finite-dimensional grid operators. In a continuous Hilbert space $\mathcal{X}$, the analogous semigroup statements require domain assumptions: a densely defined, closed, maximal dissipative operator generates a contraction C_0-semigroup, while a skew-adjoint operator generates a unitary group (Pazy, 1983; Engel & Nagel, 2000; Reed & Simon, 1972). The negative-semidefinite and skew-Hermitian symbol constraints in Section 4.2 are their finite-dimensional counterparts. Local reaction and source branches are zero-order maps; any additional sign, range, or Lipschitz condition is task dependent.

B Pseudodifferential Implementation

Pseudodifferential operators provide a symbol-level parameterization for mechanism generators. Let $x \in \mathbb{R}^d$ or $x \in \mathbb{T}^d$ denote spatial position, let $\xi \in \mathbb{R}^d$ denote the Fourier dual variable, let d denote the spatial dimension, and let $\widehat{u}(\xi)$ denote the Fourier transform of the state u. For a symbol $a(x, \xi)$, define

$$\mathrm{Op}(a)u(x) = (2\pi)^{-d} \int e^{ix\cdot\xi} a(x, \xi) \widehat{u}(\xi)\, d\xi. \tag{12}$$

Here $d\xi$ denotes integration over the frequency variable. On a finite grid, the neural network learns symbols and channel mixing on truncated frequencies. If the symbol depends only on retained Fourier modes and uses channel mixing, the result is an FNO-style spectral convolution. Allowing more general dependence on x and ξ yields a pseudodifferential neural-operator parameterization (Li et al., 2021; Shin et al., 2024). In the experimental implementation, let $\ell = 0, \ldots, L-1$ denote the layer index, let L denote the number of layers, let z_ℓ denote the hidden state at layer ℓ, let a_ℓ denote the symbol at layer ℓ, let W_ℓ denote a pointwise channel operator, let σ denote a pointwise nonlinearity, let q denote the state variable being updated, let q^+ denote the updated state after the block, and let Π denote the projection from hidden channels to state variables. A residual pseudodifferential block can be written as

$$z_{\ell+1} = \sigma(\mathrm{Op}(a_\ell) z_\ell + W_\ell z_\ell), \qquad q^+ = q + \Pi z_L, \tag{13}$$

where $\mathrm{Op}(a_\ell)$ is the symbol multiplier with channel mixing.

For a raw matrix-valued symbol $Q_m(x, \xi)$, the finite-grid constraints in Section 4.2 can be implemented by the projections

$$a_m^{\mathrm{diss}} = -Q_m^* Q_m, \qquad a_m^{\mathrm{cons}} = \frac{1}{2}(Q_m - Q_m^*). \tag{14}$$

Local reaction and source branches use pointwise channel maps and therefore introduce no spatial mixing in their leading term.

Nonuniform media are represented by multiplication operators $M_c u = cu$. If a belongs to a symbol class S^ρ of order ρ and c is sufficiently smooth, standard symbol calculus gives the commutator symbol r and a two-orders-lower remainder $r_{\rho-2}$:

$$[\mathrm{Op}(a), M_c] = \mathrm{Op}(r), \qquad r(x,\xi) = \frac{1}{\sqrt{-1}} \partial_\xi a(x,\xi) \cdot \nabla_x c(x) + r_{\rho-2}(x,\xi), \tag{15}$$

where S^ρ is the order-ρ symbol class, $r \in S^{\rho-1}$, and $r_{\rho-2} \in S^{\rho-2}$ (Hörmander, 2007; Taylor, 1981). Hence, the interaction induced by medium variation is one order lower than the principal mechanism and is controlled by ∇c. In particular, the principal commutator between a first-order mechanism and medium multiplication is zero-order. Let D_h denote a finite-difference operator with grid scale h, and let τ_h denote the corresponding grid-shift operator. The discrete product rule gives the same structure:

$$D_h(cu) = cD_h u + (D_h c)\tau_h u. \tag{16}$$

For tasks with nonuniform media, HGL represents this term through a bounded correction. Let $\tau > 0$ be its scale, let B_ψ be a lower-order bounded operator with parameters ψ, and let $g(c,u)$ be a pointwise gate. The medium-interaction map is

$$\mathcal{C}_c(u) = u + \tau\, g(c,u) \tanh\big(B_\psi(u,c)\big), \tag{17}$$

where $\tanh$ acts pointwise. With $\epsilon > 0$ and the supremum norm taken over space and channels, we use

$$g(c,u) = \frac{|\nabla c|}{\|\nabla c\|_\infty + \epsilon} \cdot \frac{\|u\|_\infty}{\|u\|_\infty + \epsilon}. \tag{18}$$

The correction vanishes for a uniform medium or a zero state.

C Steady-State Convection–Diffusion–Reaction Evaluator and Protocol

The steady-state experiment solves Equation 8. Reference solutions are computed on a 32×32 interior grid with spacing $h = 1/33$. The diffusion term uses a five-point flux discretization with harmonic averaging at cell faces, the transport term uses first-order upwind differences, and the reaction term uses pointwise multiplication. Homogeneous Dirichlet conditions are imposed by zero extension beyond the boundary. Each heterogeneous coefficient field ν is taken from the $\beta = 1$ PDEBench Darcy data (Takamoto et al., 2022). A manufactured-solution test gives relative L^2 errors of 1.15×10^{-2} and 5.97×10^{-3} on 32×32 and 64×64 grids, respectively, consistent with the first-order convergence rate of the upwind discretization.

Each of the three training equation families contains 160 training, 16 validation, and 16 test samples. In the convection–diffusion family, $|b_x|$ and $|b_y|$ are sampled independently from $[0.5, 2]$, with independent random signs. The reaction coefficient r in the screened-Poisson family is sampled from $[5, 30]$. The test set for the three-mechanism equation contains 64 samples and uses the same ranges for $\mathbf{b}$ and r. The training, validation, and test splits use disjoint ν fields. HGL's principal operator uses a five-point flux with arithmetic averaging at cell faces and centered differences for transport. A local correction learns the discrepancy from the reference discretization.

The steady-state evaluator first constructs the discrete operator in Equation 8. For each sample, let $\nu_{\min}$ and $\nu_{\max}$ denote the minimum and maximum diffusion coefficients, respectively, and define $\bar{\nu} = (\nu_{\min} + \nu_{\max})/2$. The constant-coefficient preconditioner is

$$\mathcal{P}_{\bar{\nu},\mathbf{b},r,\lambda_s} = -\bar{\nu}\Delta + \mathbf{b}\cdot\nabla + (r + \lambda_s)I, \tag{19}$$

where Δ is the discrete Laplacian, I is the identity operator, and $\lambda_s \geq 0$ is a learnable spectral shift. To apply $\mathcal{P}^{-1}$, we first remove the constant-coefficient transport term by exponential conjugation, then solve the shifted Laplacian mode by mode with a two-dimensional type-I discrete sine transform (DST-I). All training and test equation families use the same preconditioner construction.

Let u_k denote the solution at iteration k, and let $d_k = f - \mathcal{L}_{\nu,\mathbf{b},r}u_k$ denote the discrete residual. The learnable step size at iteration k is $\omega_k \in (0, 2)$, and the local-correction weight is $\gamma \in (0, 1)$. For $k \in \{0, \ldots, 23\}$, HGL applies the update

$$u_{k+1} = u_k + \omega_k \mathcal{P}^{-1} d_k + \gamma \mathcal{N}_\psi(\chi_k), \qquad \chi_k = (u_k, d_k, \mathcal{P}^{-1} d_k, \nu, \mathbf{b}, r, \nabla u_k), \tag{20}$$

where $\mathcal{N}_\psi$ is a local convolutional operator with parameters ψ, and χ_k denotes its concatenated input fields. The local output layer is initialized to zero, ω_k to 1, and λ_s to a small positive value. All mechanism compositions use the same preconditioner, iteration count, and update rule. HGL, FNO, and Transolver receive the same normalized ν, b_x, b_y, and r. The three models use the same budget of 6000 optimization steps, validation criterion, and three random seeds; their parameter counts are 407260, 463041, and 450197, respectively.

D Supplementary Experimental Details

Code and data artifact. The anonymous OSF archive contains table-specific training commands and scripts that generate synthetic trajectories or acquire external datasets. The Fisher–KPP and five-mechanism data are generated and cached by the experiment scripts. For steady CDR, the preparation script downloads the cited PDEBench Darcy fields and constructs the equation families evaluated here.

Known symbols and context-based coefficient estimation. The data lie on a 64×64 periodic grid, and each trajectory contains 96 frames. The training set contains only systems in which transport, diffusion, or reaction acts alone. Each coefficient configuration provides 128 training trajectories and 32 validation trajectories. The coefficient-interpolation and low-diffusion tests each contain 32 unseen three-mechanism trajectories. Models receive four context frames, and predictions are evaluated at steps 10, 20, 40, 60, and 80. Reported statistics aggregate three independent training runs.

HGL uses unit-strength transport, diffusion, and reaction symbols as explicit operator priors for their corresponding mechanisms in Equation 2. For context frame u_j with $j \in \{0, 1, 2, 3\}$, let $\widehat{u}_j(\kappa)$ be its Fourier coefficient at discrete frequency $\kappa = (\kappa_x, \kappa_y)$ and let Δt be the frame interval. Where $\widehat{u}_j(\kappa) \neq 0$, the fitted variant computes

$$\widehat{a}_j(\kappa) = \frac{1}{\Delta t} \log\left(\frac{\widehat{u}_{j+1}(\kappa)}{\widehat{u}_j(\kappa)} \right), \qquad j \in \{0, 1, 2\}. \tag{21}$$

Transport velocity is estimated by weighted least squares on the low-frequency imaginary part over $|\kappa_x|, |\kappa_y| \leq 40$. The diffusion coefficient is obtained by regressing the real part against $-|\kappa|^2$ over the band $30 \leq |\kappa| \leq 80$, where $|\kappa| = (\kappa_x^2 + \kappa_y^2)^{1/2}$. Let $\overline{u}_j$ denote the spatial mean of the state at frame j, and let $\overline{u_j(1 - u_j)}$ denote the spatial mean of the logistic reaction term. The reaction-coefficient estimate is

$$\widehat{\rho} = \mathrm{median}_j \, \frac{\overline{u}_{j+1} - \overline{u}_j}{\Delta t \, \overline{u_j(1 - u_j)}}. \tag{22}$$

where $\widehat{\rho}$ denotes the estimated reaction coefficient. All Fourier regressions are weighted by $|\widehat{u}_j(\kappa)|^2$. Coefficients for training, validation, and test samples are estimated from their own context frames. Let $\mathcal{A}(u)$, $\mathcal{D}(u)$, and $\mathcal{R}(u)$ denote the instantaneous tendency fields obtained by applying the transport, diffusion, and reaction operators to state u. The operator-informed baselines receive these three fields as additional input channels.

Composition and parameterization ablations. The composition ablation uses the equation-specified coefficient setting of Table 1, and all variants are trained for 200 steps. Let $K = 3$ denote the number of mechanisms, let $m \in \{1, \ldots, K\}$ index mechanisms, and let ξ denote the Fourier frequency. The operator symbol of mechanism m is $a_m(\xi)$, and its weighted symbol is $z_m(\xi) = \theta_m a_m(\xi)$. Additive HGL uses $a(\xi) = \sum_{m=1}^{K} z_m(\xi)$. The nonlinear variant uses a per-frequency MLP F_ϕ with parameters ϕ and sets $a_{\mathrm{mix}}(\xi) = F_\phi(z_1(\xi), \ldots, z_K(\xi))$. The residual nonlinear variant is

$$a_{\mathrm{res}}(\xi) = \sum_{m=1}^{K} z_m(\xi) + \delta_\phi\big(z_1(\xi), \ldots, z_K(\xi)\big), \tag{23}$$

where δ_ϕ is a zero-initialized per-frequency MLP. Both nonlinear variants share the mechanism generators, operator symbols, and evaluator with HGL. Unconstrained symbols use free complex multipliers initialized within the corresponding dissipative or skew class. Learned symbols preserve additive composition but replace the analytic principal symbols supplied by the task with complex-valued symbol fields learned from data. All remaining data and optimization settings are identical.

Robustness of context-based coefficient estimation. The robustness experiment fixes an HGL model trained with equation-specified coefficients and constructs the combined generator with $\widehat{\theta}$ at inference. Let $q \in \{2, 4, 8\}$ denote the number of context frames received by the estimator, and let $\eta \in \{0, 0.01, 0.05\}$ denote the relative noise level. We write the standard deviation of the context state u_{ctx} as $\mathrm{std}(u_{\mathrm{ctx}})$. Independent Gaussian noise with standard deviation $\eta\,\mathrm{std}(u_{\mathrm{ctx}})$ is added to the estimator input. Rollout is initialized from the uncorrupted context frames. The coefficient-identification error is

$$e_\theta = \frac{\|\widehat{\theta} - \theta\|_2}{\|\theta\|_2 + \epsilon}. \tag{24}$$

Table 7: Context and noise sensitivity for a fixed HGL model. Noise η scales with each sample's context standard deviation. Entries are median errors over 32 coefficient-interpolation test trajectories; bold marks the minimum across context lengths within each noise group.

	$\eta = 0\%$			$\eta = 1\%$			$\eta = 5\%$		
Context frames q	2	4	8	2	4	8	2	4	8
e_θ	**0.040**	**0.040**	**0.040**	**0.059**	**0.059**	0.065	**0.321**	0.402	0.431
10 steps	0.00261	0.00259	**0.00253**	0.00554	**0.00487**	0.00504	0.0274	**0.0257**	0.0290
80 steps	0.00919	0.00912	**0.00895**	0.0188	0.0147	**0.0140**	0.0878	0.0650	**0.0649**

Partial-combination coverage. The five-mechanism experiment uses four context frames to predict the subsequent trajectory and evaluates steps 10, 20, 40, and 60. Each mechanism composition contains two coefficient settings, and each setting provides four training trajectories and one validation trajectory. When training systems contain at most one, two, three, or four mechanisms, all models are trained for 200, 600, 1000, and 1200 steps, respectively. Each setting uses three independent training runs. HGL uses the same parameterization under all four coverage levels: linear transport and diffusion use known analytic symbols, while nonlinear advection, reaction, and source terms are learned from data. The gated correction is bounded by 0.10.

For each table entry, we first take the median over test samples and then the median over three independent training runs. All models use the same data splits and prediction horizons.

Tables 8, 9, 10, and 11 report complete results under the four training-coverage levels. HGL achieves the lowest error at every coverage level and prediction horizon.

Table 8: Relative L^2 errors on the unseen five-mechanism system when training systems contain at most one mechanism. Parameter counts are in millions.

Model	Params	10 steps	20 steps	40 steps	60 steps
HGL	1.60	**0.096**	**0.174**	**0.299**	**0.385**
U-Net	1.63	0.272	0.456	0.639	0.686
DeepONet	1.61	0.251	0.434	0.627	0.697
DPOT	1.64	0.261	0.434	0.616	0.675
CNO	1.65	0.266	0.434	0.621	0.677
FNO	1.64	0.263	0.436	0.616	0.662
Galerkin Transformer	1.67	0.265	0.432	0.621	0.717
Transolver	1.62	0.267	0.428	0.603	0.684

Table 9: Relative L^2 errors on the unseen five-mechanism system when training systems contain at most two mechanisms. Parameter counts are in millions.

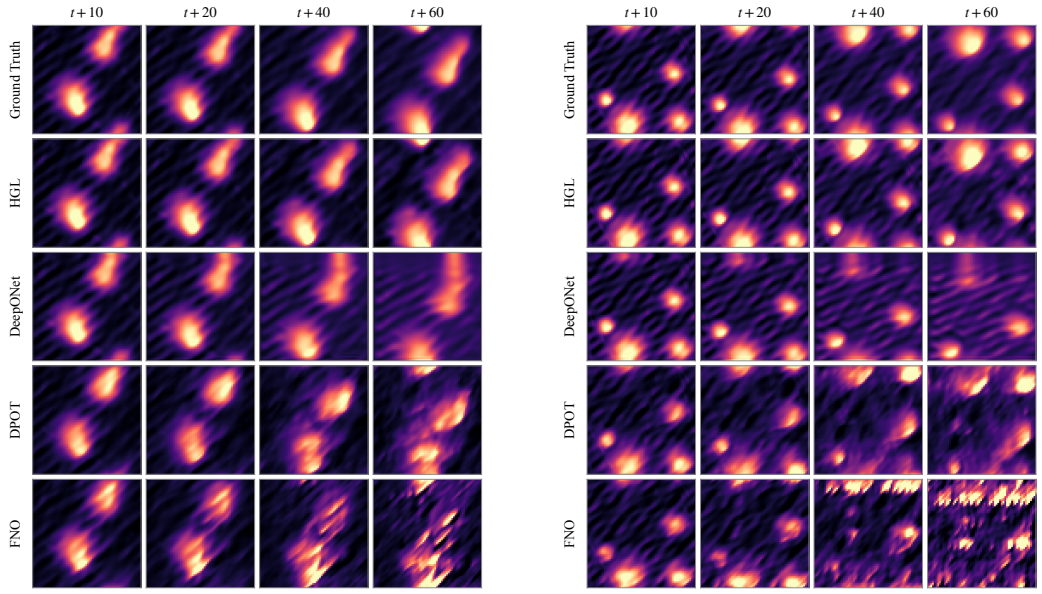
Model	Params	10 steps	20 steps	40 steps	60 steps
HGL	1.60	**0.056**	**0.110**	**0.216**	**0.318**
U-Net	1.63	0.120	0.214	0.368	0.493
DeepONet	1.61	0.127	0.246	0.450	0.614
DPOT	1.64	0.151	0.242	0.357	0.429
CNO	1.65	0.123	0.222	0.371	0.463
FNO	1.64	0.155	0.291	0.540	0.936
Galerkin Transformer	1.67	0.189	0.363	0.572	0.657
Transolver	1.62	0.209	0.398	0.566	0.585

Table 10: Relative L^2 errors on the unseen five-mechanism system when training systems contain at most three mechanisms. Parameter counts are in millions.

Model	Params	10 steps	20 steps	40 steps	60 steps
HGL	1.60	**0.037**	**0.067**	**0.124**	**0.179**
U-Net	1.63	0.087	0.165	0.302	0.416
DeepONet	1.61	0.089	0.177	0.307	0.419
DPOT	1.64	0.137	0.226	0.331	0.389
CNO	1.65	0.100	0.191	0.353	0.528
FNO	1.64	0.182	0.315	0.557	0.876
Galerkin Transformer	1.67	0.177	0.368	0.571	0.599
Transolver	1.62	0.188	0.399	0.567	0.584

Table 11: Relative L^2 errors on the unseen five-mechanism system when training systems contain at most four mechanisms. Parameter counts are in millions.

Model	Params	10 steps	20 steps	40 steps	60 steps
HGL	1.60	**0.035**	**0.069**	**0.128**	**0.193**
U-Net	1.63	0.078	0.152	0.270	0.361
DeepONet	1.61	0.067	0.135	0.234	0.308
DPOT	1.64	0.139	0.204	0.292	0.350
CNO	1.65	0.069	0.127	0.229	0.350
FNO	1.64	0.185	0.323	0.574	0.906
Galerkin Transformer	1.67	0.193	0.385	0.547	0.585
Transolver	1.62	0.188	0.401	0.591	0.593



(a) Additional test trajectory I.

(b) Additional test trajectory II.

Figure 3: Additional long-horizon predictions on the unseen five-mechanism composition. Training data cover compositions of up to four mechanisms, and the figure compares predicted fields at steps 10, 20, 40, and 60. Ground truth and model predictions use a shared color scale within each trajectory.

E Extended Evaluations

E.1 Darcy Source-Strength Interpolation

We use the PDEBench Darcy equation for a static source-strength interpolation task (Takamoto et al., 2022). The elliptic operator is fixed while the source strength varies between two training values. The equation is defined on $\Omega = (0,1)^2$:

$$-\nabla\cdot\big(\nu(x)\nabla u(x)\big) = \beta, \qquad u|_{\partial\Omega} = 0, \tag{25}$$

Here $x \in \Omega$ is the spatial coordinate and $\partial\Omega$ is the domain boundary. The coefficient field is $\nu : \Omega \to \mathbb{R}_+$, the steady-state solution is $u : \Omega \to \mathbb{R}$, and the constant source strength satisfies $\beta > 0$. Let $G_\nu u = \nabla\cdot(\nu\nabla u)$ denote the elliptic generator. HGL uses the analytic principal symbol of G_ν as an operator prior and approximates $(-G_\nu)^{-1}$ by preconditioned iteration.

Training strengths are $\beta \in \{0.01, 10\}$, and the unseen interpolation strengths are $\beta \in \{0.1, 1.0\}$. Each value of β has 320 samples. Each training strength provides 128 training samples and 32 validation samples, while each test strength provides 64 test samples. Models are trained for at most 1000 steps, with checkpoints selected by validation error. Input coefficient fields and output solution fields are normalized using training-set statistics.

Table 12: Normalized relative L^2 errors for Darcy source-strength interpolation. Each entry reports the median and range $[\min, \max]$ over three independent training runs, and parameter counts are in millions. The lowest median in each column is bold.

Model	Params	$\beta = 0.1$	$\beta = 1.0$
HGL	1.08	**0.035** $[0.028, 0.051]$	**0.308** $[0.292, 0.833]$
FNO	1.18	0.332 $[0.299, 0.344]$	0.745 $[0.620, 0.834]$
Transolver	1.09	0.051 $[0.050, 0.063]$	0.967 $[0.859, 1.578]$

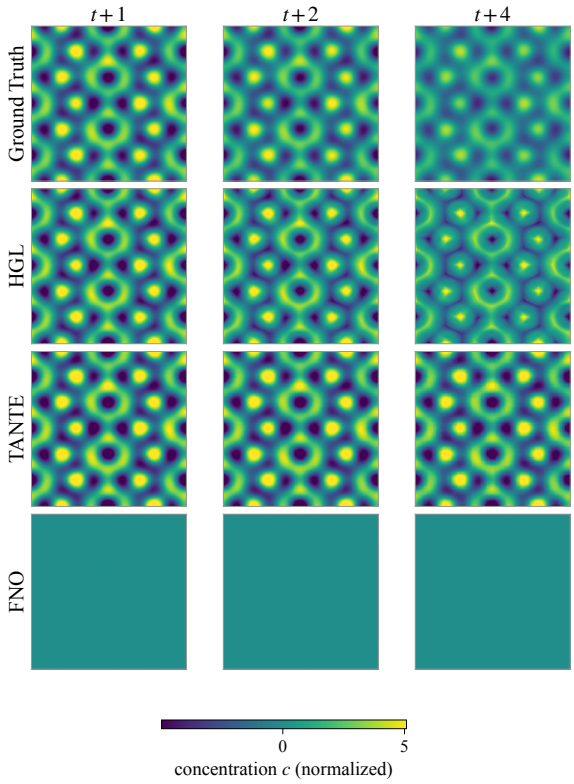
HGL achieves the lowest median error at both unseen source strengths. The corresponding steady-state predictions are shown in Figure 4b.

E.2 Multi-Field Dynamics in The Well

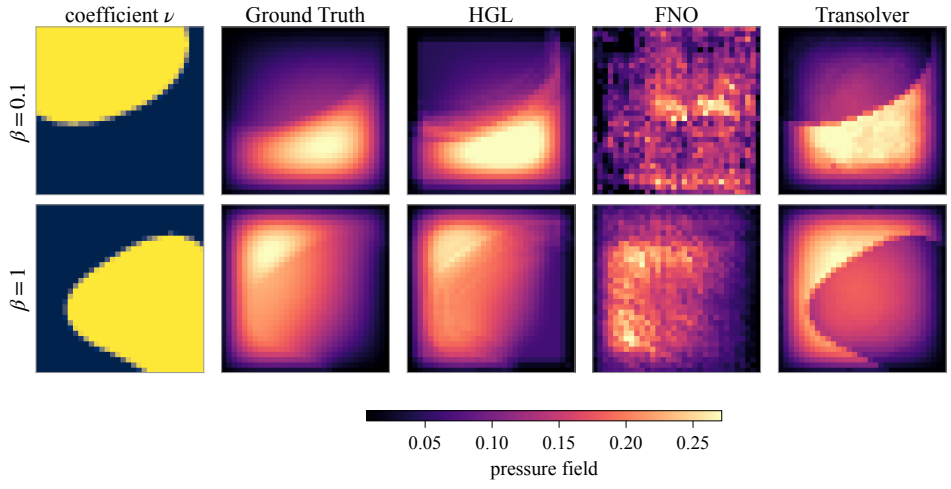
The Well (Ohana et al., 2024) provides three multi-field systems without analytic mechanism symbols: active matter, the two-dimensional turbulent radiative layer, and the Helmholtz staircase. The generator representation is therefore learned entirely from data. Evaluation includes standard trajectory validation, parameter interpolation within the training range, and parameter extrapolation beyond the training range. We compare against TANTE (Wu et al., 2026), AFNO (Guibas et al., 2022), FNO, and a convolutional U-Net.

All models receive four time steps and predict the next eight, with batch size one. In active matter, α denotes active dipole strength and ζ denotes orientational alignment strength. Training combinations use $\alpha \in \{-1, -3, -5\}$ and $\zeta \in \{1, 5, 9, 13\}$; parameter interpolation uses the remaining combinations with $\zeta \leq 13$, and parameter extrapolation uses $\zeta \in \{15, 17\}$. In the turbulent radiative layer, the training, interpolation, and extrapolation values of cooling time t_{cool} are $\{0.03, 0.1, 0.32, 1.0\}$, $\{0.06, 0.18, 0.56\}$, and $\{1.78, 3.16\}$, respectively. In the Helmholtz staircase, the training, interpolation, and extrapolation values of acoustic angular frequency ω are $\{2, 6, 9, 13, 17, 20\}$, $\{4, 8, 11, 15, 18, 21\}$, and $\{23, 24, 25\}$, respectively. The models are trained for 12, 20, and 4 epochs on the three datasets, respectively, with checkpoints selected by validation error.

Concentration predictions for active-matter parameter interpolation are shown in Figure 4a.



(a) Concentration predictions for active-matter parameter interpolation.



(b) Steady-state predictions for Darcy source-strength interpolation.

Figure 4: Qualitative predictions in the extended evaluations. (a) Ground truth and model predictions for active matter use a shared color scale. (b) Each Darcy row shares the color scale of its ground-truth field.